

DGtal: Volumetric Geometry Package

<http://liris.cnrs.fr/dgtal>

D. Coeurjolly

Package description

Should contain

- Methods performing geometric analysis of images, sets or objects as subset of \mathbb{Z}^d
- $\mathbb{Z}^d \rightarrow \mathbb{Z}^d$ functions

Examples

- Distance transformation, Reverse Distance, Digital Medial Axis extraction
- Geometrical moments computation
- Global volumetric shape descriptors
- ...
- Image transformation ? (Quasi-Affine Transform, digital rotations,...)

Location

- `{DGtal}\src\DGtal\geometry\ND\volumetric`
- `{DGtal}\tests\DGtal\geometry\ND`

In DGtal 0.4

Available:

- *dD* Separable Distance Transform (l_1, l_∞, l_2)
- *dD* Reverse DT (l_1, l_2)
- *dD* Simple measure (area, volume,...) shape descriptor

In progress (github branch):

- Digital Voronoi mapping

Scheduled:

- Medial axis extraction

Separable Distance Transformation

For each point of an object, we compute the minimum distance to the background

Overview of the algorithm

- Separable decomposition of the metric and the minimization process
- for each dimension, we have a double-scan of the volume

⇒ $O(d \cdot n^d)$ for a n^d image.

Which metric?

- Any weighted l_p metric
- Chamfer mask in 2D
- ...

⇒ `SeparableMetricTraits`

Bottleneck

- For exact computations, the range of the output image value type is $O(d \cdot n^p)$.
- In the current implementation, we have a *double buffering* of the output image (could be replace to a single 1D buffer)

Implementation

DistanceTransformation

- Parametrized by an input image type, a static “p” value, and an optional internal value type
- Defines an `OutputImage` type
- Main method:

```
template <typename ForegroundPredicate>  
OutputImage compute(const Image & inputImage, const ForegroundPredicate  
& predicate );
```

ReverseDistanceTransformation

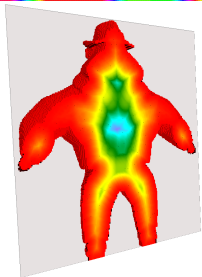
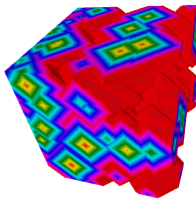
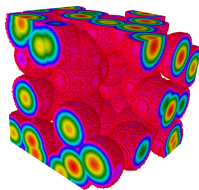
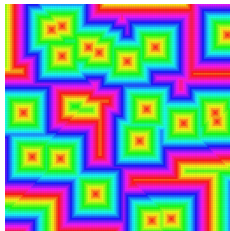
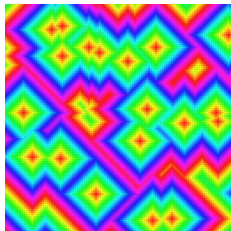
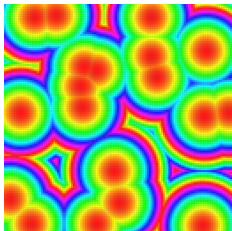
- Parametrized by an input image type, a static “p” value, and an optional internal value type
- Defines an `OutputImage` type
- The constructor needs two values for the background/foreground of the reconstruction
- Main methods:

```
OutputImage reconstruction(const Image & inputImage);  
  
template<typename DigitalSet>  
void reconstructionAsSet(DigitalSet &aSet, const Image &inputImage);
```

Usage

```
1      //Domain BBox
2      Z2i::Point a ( 0, 0 );
3      Z2i::Point b ( 127, 127);
4
5      //Input image with unsigned char values
6      typedef ImageSelector<Z2i::Domain, unsigned int>::Type Image;
7      Image image ( a, b );
8
9      //We fill the image with the 128 value
10     for ( Image::Iterator it = image.begin(), itend = image.end();it != itend; ++it)
11     image.setValue(it)=128;
12     //We generate 16 seeds with 0 values.
13     randomSeeds(image,50,0);
14
15     //Types
16     typedef DistanceTransformation<Image, 2> DTL2;
17     typedef DistanceTransformation<Image, 0> DTLInf;
18
19     DTL2 dtL2;
20     DTLInf dtLinf;
21
22     //Main Computation
23     DTL2::OutputImage resultL2 = dtL2.compute ( image );
24     DTLInf::OutputImage resultLinf = dtLinf.compute ( image );
25
26     //Reconstruction types for the 12 metric
27     typedef ReverseDistanceTransformation< DTL2::OutputImage, 2 > ReverseDTL2
28     typedef ReverseDTL2::OutputImage ImageRDT;
29     ReverseDTL2 reverseDT;
30
31     //REDT Computation
32     ImageRDT reconstruction = reverseDT.reconstruction( resultL2 );
```

Examples



Future Works

TODO list

- Voronoi/Power diagram mapping (OutputImage = ImageContainer<Point>)
- RMA extraction
- Benchmark/Improve memory management
- Out-of-core versions (meta tiled image container?)
- Add tools (thickness diagram, MA simplification, ...)
- Volumetric based differential estimators
- QAT

Questions

- Mimic ITK/VTK image filters for volumetric transforms: e.g. output image as a DistanceTransformation class member and we return smart pointer ?